STOCHASTIC BASED PRE-EMPTIVE PLANNING AND SCHEDULING

Peter Matthews and Graham Coates

School of Engineering, Durham University, UK
p.c.matthews@durham.ac.uk graham.coates@durham.ac.uk

Keywords: Agility, Process Simulation, Concurrent Engineering, Resource Allocation.

Abstract

This paper describes a significant revision to the Concurrent Engineering (CE) methodology that enables a shortened project completion time. Under the CE methodology, sequential tasks can only be performed as such. We introduce a method for starting sequential tasks concurrently using a pre-emptive approach. Where there are a, suitably small, finite number of possible alternative subsequent tasks, we propose that a more agile approach is to begin work on these alternative subsequent tasks concurrently to the preceding task, sharing the resource needed for the subsequent task amongst the different alternatives. Further, where the probability for each alternative task is known, we demonstrate that by setting the resource allocation equal to the probabilities of each outcome, it is possible dynamically allocate resources to minimise the expected completion of the overall project. A simple classically sequential two task case study is developed and analysed to illustrate this method. The paper concludes by revisiting the original assumptions and discussing how resource efficiency is traded off for minimising project completion time.

1 Introduction

A key competitive edge can be gained from being able to complete design and manufacture projects more rapidly than the competition. With this principle in mind, this paper introduces a methodology to support this goal and analyses the cost of achieving this aim. This paper considers how a sequential process-based system can allocate resources to decrease total overall process time. This is achieved through an 'agile' methodology that starts pre-emptively processing in parallel a set of alternative potential outcomes before the actual outcome is known. Where the probability distribution function for the various outcomes is known, the resource allocation can be optimised with respect to the expected process finish time.

Fundamentally, this approach revises the Concurrent Engineering (CE) methodology. Where CE has the distinct concepts of concurrent tasks, which can be performed in parallel, and sequential tasks, which can only be performed in sequence, the pre-emptive resource allocation approach allows for sequential tasks to be performed concurrently. A number of fields already informally adopt this principle, for example when a news editor is reporting on an election story, the journalists will be requested to submit both a "President Re-elected" and "President Defeated" version of the story *before* the election result is known. When the election outcome is officially declared, the editor selects the correct version of the story and updates any details, for example the exact poll result. This provides agility to the editor, enabling a more rapid news production process. The cost of this is efficiency: in this case the effort of writing the unprinted story is wasted.

The remainder of this paper is structured as follows: Section 2 provides the key background material that forms the basis of this work. Section 3 introduces the theoretical arguments and the problem representation. This is followed in Section 4 by a detailed case study, using simulation software to illustrate the agile pre-emptive resource allocation methodology. Finally, Sections 5 and 6 respectively discuss the case study results and conclude the paper.

2 Background

There are three main components supporting this work: Concurrent Engineering, Agile methodologies and Resource Allocation. Concurrent Engineering provides the basis of this work and Agile ideas are used to add significant revision to the CE and resource allocation methodologies.

Concurrent engineering is the distribution of the design, and potentially manufacture, work between a number of agents [Carter & Baker, 1991]. An agent in this case will be a design team, manufacturing shop, assembly facility, or some other related facility. These agents can then either recursively apply concurrent engineering again, or follow the basic linear design process if they are a 'terminal' agent. For example, the design team might sub-contract some of the design work to another specialist design group. The agents are selected according to their known expertise [Armoutis & Bal, 2003]. These agents are networked through a virtual enterprise while the project is underway and combining the work towards the end of the project. Through this network, agents communicate as necessary. This concurrent engineering approach provides a means for rapidly creating enterprises with high degrees of competency without the need to support these competencies during projects which do not require the same competency profile. Thus, the virtual enterprise has the benefits of a large, well found enterprise, without having to pay for the maintenance overhead of resources that are not required for other projects.

The concept of agility in a manufacturing context has recently emerged [Lau et al, 2003; Jiang & Fung, 2003]. Most authors agree that 'agility' is the ability to rapidly respond to some external and unexpected event. The argument promoting agility is that it enables better survival in turbulent market conditions. However, most agile responses are tailored to changes in product demand, either in the form of production levels or in alternate design. The solutions to these tend to fall in line with traditional manufacture theory (for example, by applying Just-in-Time methods) or design modification (such as mass customisation applied after the initial product launch). Thus, enterprises use the agile methods to enable them to respond to the market, based on a given design.

One of the key ideas is taken from the agile software development community [Cockburn, 2005]. Where in CE a set of tasks in sequence cannot begin until the precedent tasks have been completed, the software development approach is to pre-emptively begin work on the subsequent tasks earlier. As more information becomes available to the subsequent tasks from the preceding tasks, rework must be done. However, provided the total amount of rework plus the remaining work on the subsequent tasks does not exceed the total time it would have originally taken for the subsequent tasks, a time saving is made. This time saving results in a more rapid deployment of the finished goods. Agile methods are required for this approach, as rework introduced while the process is ongoing represents the need to be able to change the tasks with minimal impact to the overall project.

Resource allocation is an area of key significance in manufacturing [Tharumarajah, 2001; Maropoulos et al., 2003; Wallace, 2003; Wu et al., 2005], engineering/project planning [Alcaraz & Maroto, 2001; Kara et al., 2001; Leus & Herroelen, 2004; Ursu et al., 2005; Guikema, 2006] and agent systems [Chen, 2004; Galstyan et al., 2005; Li and Soh, 2005; Manvi et al., 2005; Raftopoulou et al. 2005]. In the context of manufacturing, Tharumarajah (2001) presents a survey of resource allocation methods for distributed manufacturing systems. Further, a number of issues are highlighted that are involved in distributing tasks such as how problems are decomposed, how tasks are assigned to who, and communication between resources and tasks. Similarly, Wu et al. (2005) indicate that the inherent decentralised nature of many organisations leads to investigations focussing on distributed, rather than centralised, methods for the resource allocation problem. On the theme of distributed resource allocation problem in manufacturing systems, Wallace (2003) presents a method utilising agents to manage the sequential allocation of resources. The key finding of this work was that agent-based allocation of resources utilising resource, group and actor agents has the potential to manage sequential resource allocation without encountered deadlocks. Maropoulos et al. (2003) introduce a resource model that has been developed to aid the dynamic planning of manufacturing operations within production networks.

With regard to project planning, Leus and Herroelen (2004) assert that projects involving scheduling resources often assume that perfect information is available. As this is rarely the case, a branch-and-bound algorithm is presented aimed at solving the resource allocation problem in settings where uncertainty exists and variability in task durations. On the same theme, Alcaraz and Maroto (2001) present a genetic algorithm aimed at optimising the resource allocation problem in project scheduling. Ursu at al. (2005) also present an optimisation algorithm to solve the workforce allocation problem using a distributed system of agents. The SignPosting methodology [Clarkson and Hamilton; 2000] uses the design information quality level (eg, rough estimate, first order calculation, fine grain calculation) as a means to suggest possible subsequent design tasks to be undertaken.

An approach to allocating resources to members of a concurrent design team has been presented by Guikema (2006). In the domain of concurrent engineering (CE), Kara et al. (2001) recognises that a complexity of CE in that it involves executing tasks with incomplete information. To address this complexity, a multi-project scheduling heuristic is presented, which is reported as minimising project completion time through concurrency and optimising the utilisation of resources.

3 Theoretical argument

To construct the argument for this work, it is sufficient to consider a project consisting of two sequential tasks, Task 1 followed by Task 2. This is a problem structure that the CE approach cannot address or improve upon as it requires the tasks to be performed in sequence. Under the CE methodology, the project will first complete Task 1 which provides the full information for Task 2 to be undertaken, and only then does Task 2 begin. The 'pre-emptive' approach will be to start Task 2 with incomplete information. More specifically, provided there are a sufficiently small number of distinct possible alternatives of Task 2 depending on the outcome of Task 1, the pre-emptive approach will begin work on all these alternatives concurrently. The question remains is how to optimally allocate resources to each of the Task 2 alternatives.

A number of assumptions are made regarding the tasks. Firstly, there are a finite number of alternative versions of Task 2. Realistically, it will be necessary that there are a reasonably small number of alternatives, otherwise the resources available for Task 2 are spread too thinly and insufficient work can be completed on this task. Secondly, it is assumed that at least an approximate probability distribution function (PDF) is known for the various outcomes. A naïve approximation would be to set all probabilities equal. Finally, it is assumed that the resources required for Task 1 are distinct from those required for Task 2. For example, a project might require process time on two different computers with different software packages. Effectively, it assumes that the resources of Task 2 are idle while Task 1 is underway.

3.1 Problem representation

At this point, this work only considers the time aspect of a project. The time required for Task i is denoted $t(i)$, and for simplicity the time will be represented in the reference frame of each task. Therefore, the start of each task is at time 0 with respect to that task. Clearly, the aim is to minimise the time in terms of the global reference frame. This is achieved by referencing to the earliest task, in our case Task 1.

The time where Task 1 has sufficient information to start a sufficiently small set of alternative versions of Task 2 is denoted by t_0. The probability of each alternative being the ‘correct’ outcome at the end of Task 1 is denoted p_i. Each alternative task is allocated a proportion of the total available resource for Task 2, and this is denoted by α_i.

The aim is to determine the optimal resource allocation for the given PDF in terms of expected time to complete Task 2. Once Task 1 has terminated, the ‘correct’ alternative will be known. Therefore, all of Task 2 resources can be committed to the correct, partially completed, alternative. All other alternatives are terminated at this point.

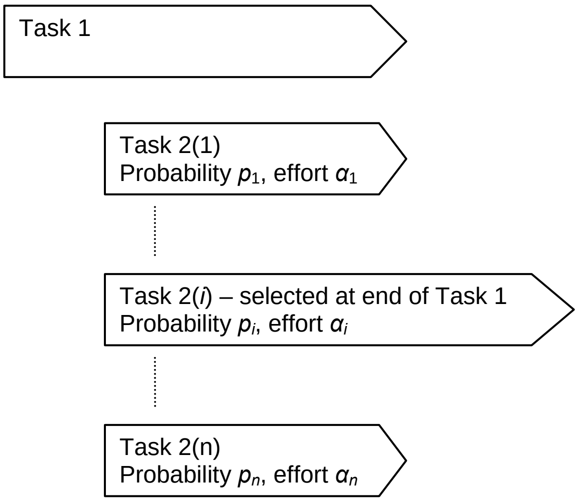

Figure 1. Temporal representation of an Agile Pre-emptive task processing for two sequential tasks

3.2 Equations: equal length alternatives

The basic case is where all alternative versions of Task 2 require the same amount of processing time, for example the same calculation is performed using different starting conditions. Figure 1 illustrates the schema of the project execution. A critical value here is the amount of time where all the alternatives are being processed, given by $t^{(1)} - t_0$, ie the time between the pre-emptive start of Task 2 and the end of Task 1. Therefore, the amount of work done on each alternative i of Task 2 is given by:

$$\alpha_i \frac{(t^{(1)} - t_0)}{t^{(2)}} \tag{1}$$

Hence, at the end of Task 1, the time remaining for alternative i is given by:

$$\begin{aligned}\Delta t_i &= t^{(2)}\left(1 - \alpha_i \frac{(t^{(1)} - t_0)}{t^{(2)}}\right) \\ &= t^{(2)} - \alpha_i (t^{(1)} - t_0)\end{aligned} \tag{2}$$

Recall, the aim is to minimise the expected total time taken to complete Task 2. Therefore, we aim to minimise:

$$\begin{aligned}\min \mathbf{E}(\Delta t) &= \sum p_i \Delta t_i \\ &= \sum p_i t^{(2)} - \sum p_i \alpha_i (t^{(1)} - t_0) \\ &= t^{(2)} - (t^{(1)} - t_0)\sum p_i \alpha_i\end{aligned} \tag{3}$$

From the above equation, it can be seen that the completion time is minimised when the second term (the remaining summation) is maximised. This summation is effectively the ‘dot product’ between the PDF vector and the resource allocation vector, and is maximised when the two vectors are parallel. As both vectors must also sum to unity, the project completion time is minimised when the resource distribution is set to the probability distribution.

3.3 Equations: varying length alternatives

The more general case is where each alternative version of Task 2 requires a different amount of processing time, namely where Task 1 determines which of a set of very different subsequent tasks would be performed. For example, a car design project where Task 1 determines what type of propulsion to use (for example either: petrochemical, electric or hybrid) and Task 2 is the design of this propulsion system. As each propulsion system in significantly different, each will require a different amount of time to complete.

The time required to complete each alternative version of Task 2 will now be denoted as $t_i^{(2)}$. Equation 3, representing the expected remaining completion time of Task 2 once Task 1 has completed is now given by:

$$\begin{aligned}\min \mathbf{E}(\Delta t) &= \sum p_i \Delta t_i \\ &= \sum p_i t_i^{(2)} - \sum p_i \alpha_i (t^{(1)} - t_0) \\ &= \sum p_i t_i^{(2)} - (t^{(1)} - t_0)\sum p_i \alpha_i\end{aligned} \tag{4}$$

This equation has the same structure as Equation 3, namely a constant term less a term dependent on the resource allocation vector. Again, the overall expression is minimised when the resource allocation distribution is equal to the PDF.

4 Case study and simulation

To illustrate the theory, a basic hypothetical car design process is described. In this case, the design process is reduced to the initial general arrangement decision and the subsequent detailing of this GA. In the first task (Task 1), the design team will determine the overall structure and characteristics of the car. In this hypothetical case, the relevant information from this task will be the vehicle's fuel type (gasoline or diesel), engine configuration (4, 6 or 8 cylinder) and driving axle (front or rear). The next task (Task 2) is to use the decision from the GA task to begin detailing further aspects of the car design. Under classical concurrent engineering procedures, this is not possible to start until the full information is available.

By using a pre-emptive approach, work is started on the downstream task before the first task has been completed. The hypothetical case study has a total of 12 possible outcomes from the first task. Initially, the downstream design team splits the effort equally between each possible outcome. As time progresses, Task 1 is able to provide greater direction on the probabilities of each outcome surviving. Task 2 then uses this information to reallocate resource with the aim of minimising their completion time once Task 1 has completed.

The case study will consider the efficiency of the process, and the cost involved. Further, it will illustrate the effect of deciding when to start Task 2 on the overall completion time. It will also illustrate the benefits in quantifiable terms of minimising the 'opportunity costs' that are made through reducing the overall project duration.

4.1 Design scenario

The hypothetical design scenario involves two tasks that each take a nominal 5 days at 100% resource utilisation to complete. Each task uses a different resource, and therefore until Task 2 is started, this resource is effectively idle. The simulation will be based on discrete days. To provide greater structure and direction to the case study, Task 1 completes sub-tasks with the following milestones:

- The driving axle is determined at the end of day 2;
- The fuel type is determined at the end of day 4; and
- The engine type is determined at the end of day 5.

This provides a rationale and direction for updating the PDF at the end of each day. After each milestone is reached and a decision on that design parameter is made, the probabilities of all solutions that are not based on this decision outcome are reduced to zero. For example, when the driving axle is set to Rear-wheel drive at the end of day 2, all design solutions with a Front-wheel design are given a zero probability of being a successful outcome. This effectively stops any subsequent work continuing on these options.

The PDF is updated at the end of each day. In addition to the updates to reflect design decisions made at Task 1 milestones, a more subjective update to each design option is also placed. This provides a means for incorporating new knowledge about the overall design status on a day to day basis. For example, if as a result of ongoing work in Task 1, there appears to be a bias towards selecting a diesel engine, the PDF can reflect this by slightly increasing the probabilities of all diesel based solutions at the cost of the gasoline based solutions. This will enable more work to be completed on the diesel solutions under the belief that this is more likely outcome.

The final outcome of Task 1 (the General Arrangement), was that the car design would be a Rear wheel drive 6-cylinder diesel engine. These decisions, along with a daily variation, are reflected in the daily updated PDF (Table 2). This PDF also includes some time slots where incorrect guesses are made regarding the final outcome. For example, during days 2 and 3 the probabilities for a gasoline-powered solution are higher than the diesel solutions. This reflects the judgement of the design state and believed outcome at these points in the design process.

The case study uses this time varying PDF as a basis for further illustration of the effect of starting Task 2 at different points. To study the trade-off between overall time saved and overall design effort expended, Task 2 is illustrated with a start time based on day 1 (representing starting Task 2 in parallel with Task 1) through to day 6 (representing starting Task 2 only upon completion of Task 1).

4.2 Implementation

The simulation was implemented using a basic spreadsheet model. There are two core components to this model: the time varying PDF, representing the PDF snapshot for each discrete time segment; and the work completion status, representing how much work has been completed on each design alternative.

The PDF represents the probability of each design alternative being successful at the end of Task 1, and is used to determine how much effort should be expended on each alternative. This PDF therefore drives how much work is completed during each time segment. The work completed is a monotonically increasing count of the proportion of work completed on each alternative. For any given time segment, it is given by the work completed in the previous time segment plus the amount of work that is available to that alternative for that time segment, unless this sum is greater than 100%, in which case it is capped at 100%. Clearly, when a design option is no longer viable (i.e., its probability has collapsed to zero), the work completed level remains constant from that point onward.

For this case study it was sufficient to only model the work completed on Task 2. Task 1 is assumed to proceed in the background, only supplying information about its status at the end of each day to update the PDF.

The time and effort analysis considered two aspects: the total effort expended on Task 2 and the total amount of time saved. Computing the total amount of work expended on Task 2 simply required summing the amount of work completed at the end of the last time segment on all alternatives. To compute the time saved with greater accuracy than given by the discretised time segments, the exact completion time was calculated within the final discrete time segment.

4.3 Results

The overall results illustrate that clear time savings are to be made. Table XXX reports the overall time saving and effort expended for this case study for the various starting points

T2 start (day)	Total Effort (%)	Time Saved (%)
1	178	21.8
2	160	20.2
3	142	18.0
4	125	15.0
5	110	10.0
6	100	0

Table 1. Effort v. Time saved for the various Task 2 starting points.

As can be seen from the overall results, the time saving is considerable. In this case, as Task 2 was discretised over five days, the only visible time saving is visible where Task 2 started on either day 1 or 2. However, it can also be seen that these time savings are made at considerable expense. The greatest time saving nearly doubles the cost in terms of effort invested. One reason for this is the large number of alternatives being considered for the earlier Task 2 start times (i.e. 12 and 11 alternatives on days 1 and 2, respectively). This resulted in little work being accomplished on the eventually successful alternative during these early days.

4.4 Discussion on Opportunity Costs

It is clear that adopting a pre-emptive development methodology invokes a significant cost for what could be argued to be a relatively small benefit. However, where opportunity costs can be quantified, this enables an objective debate to be held. The opportunity costs represent the potential loss of earnings due to not being able to invest time in investigating new opportunities.

To analyse this, a simple set of graphs are plotted. These consider the difference between the effort invested less the value of the opportunity cost saved by a pre-emptive start. The graphs plot the difference for a range of ratios of effort cost versus opportunity cost. In Figure 2 three cases are considered: starting the work on Task 2 at the same time as Task 1, starting Task 2 on day 4 and starting Task 2 on completion of Task 1 (day 6), i.e. the sequential case. The horizontal axis represents the varying opportunity:effort ratio, starting at 1:5 (0.2: opportunity is worth 20% of effort expended) ranging through to 5:1 (5.0: opportunity is worth five times effort expended). The sequential case can be used as the datum point for comparison. Where the pre-emptive line intersects the datum line identifies the opportunity to cost ratio at which the pre-emptive approach (for that starting point) is of equal value to the sequential approach. Where the total cost (relative to the datum) is above 1.0, the pre-emptive approach is worse. Where the total cost is below 1.0, the pre-emptive approach is arguably better, and provides a more agile environment to the sequential approach. From the graph, it can be seen that this occurs for relatively low opportunity:effort ratios, and that therefore this is likely to be beneficial in most cases where opportunity is quantified as a valuable commodity.

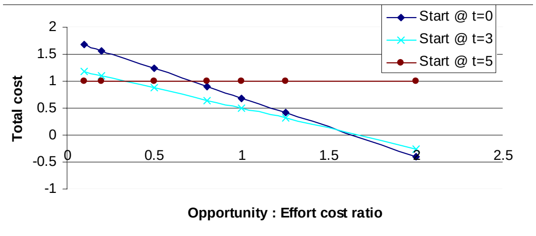

Figure 2. Cost difference between effort invested less opportunity costs, displayed for two pre-emptive start cases and the non-pre-emptive case.

4.5 PDF generating support tool

An important aspect of the pre-emptive theory lies within the ability of generating ‘good’ estimates of the PDF for the various task outcomes. The challenge herein is the general difficulty in estimating these probabilities where little prior evidence is available. Further, human decision makers in general find it difficult to place exact probability values on various outcomes. Hence, a support tool is needed to guide designers towards reasonable probability estimates.

The PDF generating support tool initially uses the naïve estimate that all outcomes are equally likely. Next, the design team can rank the potential outcomes. Finally, a rough gradient needs to be placed on this ranking. This is achieved by providing relative outcomes between a subset of the alternatives (e.g. alternative 1 is twice as likely an outcome than alternative 4). Initially, a linear gradient is used between these estimated fixed points. This overall shape can then be adapted as the designer sees fit. Finally, as the PDF must sum to unity, the designer based profile can be transformed into an estimated PDF.

5 Discussion

As a result of running the series of simulations, an interesting graph can be plotted (see Figure 4). The shape of this graph shows that the earlier Task 2 is pre-emptively started, the greater the proportion of waste. Further, this is not a linear relationship. This is to be expected, as clearly the later Task 2 is started the amount of work ‘wasted’ disappears to zero when Task 2 is started on completion of Task 1. At the other extreme, the earlier Task 2 is started will always result in some non-zero amount of Task 2 being usefully completed on completion of Task 1. As can be seen from the graph, the efficiency has an initial rapid rise, after which there is a diminishing rate of return for further delaying the pre-emptive start. However, this graph does not provide a clear point to an optimal time to start the pre-emptive work. Instead, under this case study, it remains a largely subjective decision as to how much wasted work is acceptable.

A further point to note with the presented case study regards the realism of being able to perfectly subdivide the resource for Task 2 arbitrarily, and with no context switching costs. This might be feasible where Task 2 is a strict computational job, and context switching is relatively inexpensive or where the resources available for Task 2 are in such abundance (e.g. a project team of 100 people) that this is possible. A more realistic scenario should also include overheads that are incurred with each alternative task.

Finally, this case study defined the PDF of all the alternative tasks and used this data to allocate resources. Within a real scenario this is an unlikely condition. This raises the question of how sensitive the pre-emptive system is to the accuracy of the estimated PDF.

6 Conclusions

The case study demonstrated that an overall reduction in project completion time can be achieved using pre-emptive resource allocation. This pre-emptive approach provides a level of agility to a project, however it comes at a cost of overall efficiency. Under this scheme, a proportion of resources are used to generate wasted work. Under the assumptions, this did not matter as the key objective was to minimise the overall project completion time. The approach taken in this paper could easily be adapted to consider more complex and complete objective functions. These objectives should include other costs incurred, and therefore partially wasted, by starting the subsequent task early.

Future work will also need to revisit the other assumptions. It was assumed that the PDF between the various alternative subsequent tasks remained static with time. This is unlikely, and a more typical scenario will be one where while the primary task completes a continuously improving image of the PDF becomes available, and therefore a dynamic PDF should be considered. As a result, what is the optimal resource allocation strategy under these dynamic conditions? Further links should be made to the occurrence of unexpected (external) events, thereby increasing the agility of this approach.

One potential means for addressing the dynamic nature of the PDF is to consider the iterative nature of the design process. This provides a means for updating the PDF with each iteration of the first task. Under the 'traditional' CE perspective, the subsequent task is only started once sufficient iterations of the first task have been undertaken to provide sufficiently high quality starting conditions for the subsequent task.

Finally, it was assumed that the PDF of the various alternatives was readily available. It will be necessary to conduct field research to measure how realistic an assumption this is, and to develop methods for approximating the shape of the PDF.

References

[1] Alcaraz, J. and Maroto, C. A Robust Algorithm for Resource Allocation in Project Scheduling. *Annals of Operations Research*, 102(1/4), 83-109, 2001.

[2] Armoutis, ND and J Bal. Building the Knowledge Economy: Issues, Applications, and Case Studies, chapter *E-Business through Competence Profiling*, pages 474-482. IOS Press, 2003.

[3] Carter, DE and B S Baker. Concurrent Engineering: The Product Development Environment for the 1990s. Addison-Wesley, 1991.

[4] Chen, Z-L. Simultaneous Job Scheduling and Resource Allocation on Parallel Machines. *Annals of Operations Research*, 129(1-4), 135-153, 2004.

[5] Clarkson, P.J. and Hamilton, J.R. Signposting: a parameter-driven task-based model of the design process. *Research in Engineering Design*, 12(1), 18-38, 2000.

[6] Cockburn, A. Two case studies motivating efficiency as a "spendable" quantity, *Proceedings of the International Conference on Agility* (ICAM2005), 1-6 2005.

[7] Galstyan, A., Czajkowski, K. and Lerman, K. Resource Allocation in the Grid with Learning Agents. *Journal of Grid Computing*, 3(1-2), 91-100, 2005.

[8] Guikema, S. Incentive compatible resource allocation in concurrent design. *Engineering Optimization*, 38(2), 209-226, 2006.

[9] Jiang, Z and R Y K Fung. An adaptive agile manufacturing control infrastructure based on TOPNs-CS modelling. *International Journal of Advanced Manufacturing Technology*, 22:191-215, 2003.

[10] Kara, S. Kayis, B. and Kaebernick, H. Concurrent Resource Allocation (CRA): A Heuristic for Multi-Project Scheduling with Resource Constraints in Concurrent Engineering. *International Journal of Concurrent Engineering: Research and Applications*, 9(1), 64-73, 2001.

[11] Lau, HCW, C W Y Wong, K F Pun, and K S Chin. Virtual agent modelling of an agile supply chain infrastructure. *Management Decision*, 41(7):625-634, 2003.

[12] Li, X. and Soh, L-K. Hybrid negotiation for resource coordination in multiagent systems. *Web Intelligence and Agent Systems: An International Journal*, 3(4), 231-259, 2005.

[13] Lues, R. and Herroelen, W. Stability and resource allocation in project planning. *IIE Transactions*, 36(7), 667-682, 2004.

[14] Manvi, S.S., Birje, M.N. and Prasad, B. An Agent-based Resource Allocation Model for Computational Grids. *Multiagent and Grid Systems*, 1(1), 17-27, 2005.

[15] Maropoulos, P.G., Bramall, D.G., McKay, K.R., Rogers, B. and Chapman, P. An aggregate resource model for the provision of dynamic resource-aware planning. *Proc. IMechE Part B Journal of Engineering Manufacture*, 217(10), 1471-1480, 2003.

[16] Raftopoulou, P., Koubarakis, M., Stergiou, K. and Triantafillou, P. Fair Resource Allocation in a Simple Multi-Agent Setting. *International Journal of Artificial Intelligence Tools*, 14(6), 887-899, 2005002E

[17] Tharumarajah, A. Survey of resource allocation methods for distributed manufacturing systems. *Production Planning and Control*, 12(1), 58-68, 2001.

[18] Ursu, M.F., Virginas, B., Owusu, G. and Voudouris, C. Distributed resource allocation via local choices: A case study of workforce allocation. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 9(4), 293-301, 2005.

[19] Wallace, A. Sequential resource allocation utilizing agents. *International Journal of Production Research*, 41(11), 2481-2499, 2003.

[20] Wu, T. Ye, N. and Zhang, D. Comparison of distributed methods for resource allocation. *International Journal of Production Research*, 43(3), 515-536, 2005.

Appendix

This appendix contains the PDF used in the case study simulation scenario. Design decisions are made at the end of Day 2 (drive wheel is set to Rear), Day 3 (Fuel type is set to diesel) and Day 5 (a six cylinder engine is selected). These decisions can be seen through the PDF collapsing to zero for these design options in Table 2.

Tables 3-5 display the total amount of effort expended on each design alternative for different Task 2 starting scenarios.

	Daily PDF									
Design:	1	2	3	4	5	6	7	8	9	10
FG4	0.08	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FG6	0.08	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FG8	0.08	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD4	0.08	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD6	0.08	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD8	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RG4	0.08	0.11	0.15	0.06	0.00	0.00	0.00	0.00	0.00	0.00
RG6	0.08	0.07	0.20	0.25	0.00	0.00	0.00	0.00	0.00	0.00
RG8	0.08	0.11	0.20	0.19	0.00	0.00	0.00	0.00	0.00	0.00
RD4	0.08	0.07	0.10	0.06	0.13	0.00	0.00	0.00	0.00	0.00
RD6	0.08	0.11	0.15	0.25	0.50	1.00	1.00	1.00	1.00	1.00
RD8	0.08	0.11	0.20	0.19	0.38	0.00	0.00	0.00	0.00	0.00

Table 2. Daily probabilities of each design alternative being successful. At the outset, all design alternatives are equally likely (p=0.08). (Code: DRIVE-AXLE:FUEL:ENGINE, thus FG4 is Front wheel gasoline 4 cylinder).

	Work completed on each option				(T2 start @ t=0)					
Design:	1	2	3	4	5	6	7	8	9	10
FG4	0.02	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
FG6	0.02	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
FG8	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
FD4	0.02	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
FD6	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
FD8	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
RG4	0.02	0.04	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.08
RG6	0.02	0.03	0.07	0.12	0.12	0.12	0.12	0.12	0.12	0.12
RG8	0.02	0.04	0.08	0.12	0.12	0.12	0.12	0.12	0.12	0.12
RD4	0.02	0.03	0.05	0.06	0.09	0.09	0.09	0.09	0.09	0.09
RD6	0.02	0.04	0.07	0.12	0.22	0.42	0.62	0.82	1.00	1.00
RD8	0.02	0.04	0.08	0.12	0.19	0.19	0.19	0.19	0.19	0.19

Table 3. Amount of work expended on each design alternative for starting Task 2 at the same time as Task 1, ie t=0.

	Work completed on each option			(T2 start @ t=3)						
Design:	1	2	3	4	5	6	7	8	9	10
FG4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FG6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FG8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RG4	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
RG6	0.00	0.00	0.00	0.05	0.05	0.05	0.05	0.05	0.05	0.05
RG8	0.00	0.00	0.00	0.04	0.04	0.04	0.04	0.04	0.04	0.04
RD4	0.00	0.00	0.00	0.01	0.04	0.04	0.04	0.04	0.04	0.04
RD6	0.00	0.00	0.00	0.05	0.15	0.35	0.55	0.75	0.95	1.00
RD8	0.00	0.00	0.00	0.04	0.11	0.11	0.11	0.11	0.11	0.11

Table 4. Amount of work expended on each design alternative for starting Task 2 on Day 3 (ie t=4).

	Work completed on each option			(T2 start @ t=5)						
Design:	1	2	3	4	5	6	7	8	9	10
FG4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FG6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FG8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FD8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RG4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RG6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RG8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RD4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RD6	0.00	0.00	0.00	0.00	0.00	0.20	0.40	0.60	0.80	1.00
RD8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5. Amount of work expended on each design alternative for starting Task 2 on completion of Task 1 (ie t=6).